Attorney Docket No. 03242.P008 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

MULTIPLE MEMORY ALIASING FOR A CONFIGURABLE SYSTEM-ON-CHIP

Inventors:
Jean-Didier Allegrucci
Jerry Case

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL 617 210 251 US

Date of Deposit December 22, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Tina Domingo
(Typed or printed name of person mailing paper or fee)

12-22-2000
(Signature of person mailing paper or fee) Date

MULTIPLE MEMORY ALIASING FOR A CONFIGURABLE SYSTEM-ON-CHIP

FIELD OF INVENTION

The present invention is related to a system on a chip.

BACKGROUND OF THE INVENTION

In a typical system on a chip system, the memory map at reset is different than the memory map used by the application. In general, a slow external non-volatile memory such as EPROM or FLASH is mapped at the bottom of the memory at reset. It contains the application setup code such as an RTOS for example. Upon reset, the CPU starts executing at address 0 in the external non-volatile memory and sets-up the application. Part of the setup code might place the interrupt service routines and other critical sections of the codes into a fast memory (internal RAM or fast external). Once the application setup is completed, the mapping of the non-volatile memory at the bottom of the memory is disabled. At that point, the fast memory is mapped at address 0.

SUMMARY OF THE INVENTION

A method for multiple memory aliasing for a configurable system-on-a-chip, including executing code from an internal memory, locating a configuration program in the internal memory, disabling the internal memory alias, and jumping to a secondary initialization routine, is disclosed.

These features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

Figure 1 shows an embodiment of a system on a chip.

Figure 2 shows an embodiment of a method for multiple memory aliasing.

Figures 3A, 3B and **3C** show states of memory map during configuration of a system on a chip using the method of **Figure 2.**

DETAILED DESCRIPTION

A method for multiple memory aliasing for a configurable system-on-a-chip, including executing code from an external memory, locating a configuration program in the internal memory, disabling the internal memory alias, and jumping to a secondary initialization routine, is disclosed. The multiple memory aliasing for a configurable system-on-chip enables configuration of a system on a chip without impacting the system.

As shown in **Figure 1**, the system on a chip integrates, on a single device, an embedded microprocessor 105, internal static RAM (cache + searchpad) 110, a high-speed dedicated system bus 115, and configurable logic 120, which are connected to the microprocessor 105 and system bus 115. The result is a high performance, integrated, fully static single chip system for embedded systems applications. In one embodiment, the microprocessor 105 is a general-purpose 32-bit microprocessor, which supports the standard ARM 32-bit instruction set and reduced 16-bit instruction set.

The microprocessor 105 is paired with a memory 110, which may include 8Kbyes unified code/data cache and 16Kbytes scratch pad for storage of timing critical code or data structure, for example. The cache and additional memories may be used to improve the overall performance of the system.

The configurable logic 120, may be an embedded SRAM-based Configurable System Logic (CSL) matrix, for example. The configurable logic

120 can provide "derivative on demand" system customization. The high-performance configurable logic architecture 120 may include a highly interconnected matrix of CSL cells, for example. Resources within the matrix provide easy, seamless access to and from the internal system bus 115. Each CSL cell performs various potential functions, including combinatorial and sequential logic. The combinatorial portion performs Boolean logic operations, arithmetic functions, and memory. The sequential element performs independently or in tandem with the combinatorial function. The abundant programmable input/output blocks (PIOs) 125 provide the interface between external functions and the internal system bus or configurable system logic. Each PIO offers advanced I/O options including selectable output drive current, optional input hysteresis, and programmable low-power functionality during power-down mode.

A high-performance internal system bus 115, which may be a Configurable System Interconnect (CSI) bus, for example, interconnects the microcontroller 105, its peripherals 130, and the CSL matrix 120 at a speed of 66 MHz, for example. In one embodiment, the bus 115 provides 32 bits of read data, 32 bits of write data, and a 32-bit address. Multiple masters arbitrate for bus access. Potential bus masters include the microcontroller 105, the JTAG interface 135, the read and write channels of each DMA channel, and the memory interface unit 140 (MSSIU) in some modes of operation. Functions

implemented in the CSL matrix 120 can use a DMA channel as a "proxy" master, re-using the control logic already contained in the DMA channels to become a master on the CSI bus 115. The CSI, as well as the local CPU bus 145, may follow the little endian format.

A static memory interface unit (part of the memory sub-system MSSIU 140) connects the configurable processor 120 to external static memories of the type FLASH or SRAM. The MSSIU 140 can connect to an external FLASH memory device that holds an initialization program plus the user's code. The MSSIU interface is reusable for connections to other external components. The external read, write, control, and chip-select signals are programmable providing flexible set-up, strobe, and hold timing. A direct connection from the CPU 105 to the MSSIU 140 (bypass mode) is provided to remove any additional latency incurred going through the CSI 120. For low frequency applications, the microprocessor 105 can fetch its instructions from the external FLASH directly. Assuming a FLASH with a one cycle access time, non-sequential accesses from the microprocessor incur one wait state, while sequential accesses incur no wait states.

Support for external dynamic memories provides the user with some of adding affordable and dense memories to the system. A SDRAM interface unit (part of MSSIU 140) connects the system interconnect to a variety of SDRAM types and configurations. It also supports 100-pin DIMMs. The interface can

runs at 66MHz, for example, and provides ways to optimize the interface timing for slower clocks. SDRAMs can be used as temporary memory mostly for DMA data buffering. A direct connection from the CPU 105 to the SDRAM interface 140 is also provided to remove additional CSI latency.

The pins may be shared between the FLASH and SDRAM interfaces. To provide maximum use of the CSI bandwidth and external memory bus, the cache should be enabled. A cache miss results in the CPU using the external memory bus. Minimizing the miss rate allows the higher use of the external memory bus for DMA data movement to and from SDRAM.

The four-channel DMA controller 145 provides high-bandwidth communication between functions. Its easy-to-use handshake simplifies interface and control logic. Functions from within the CSL matrix can request DMA transfers, the DMA controller providing "proxy" bus mastering capabilities. Each DMA channel can be controlled directly by the CPU or through a descriptor table that can reside anywhere within the system memory.

A set of common dedicated peripherals 130 is available. In one embodiment, the set includes 2 16-bit with pre-scalers, 2 serial controllers, a watchdog timer and an interrupt controller.

The majority of the system, including the microcontroller, can operate from a single bus clock signal. Optional sources for the bus clock include driving it directly from an off-chip signal, connecting an external crystal or

ceramic oscillator between the dedicated crystal-oscillator amplifier pins and synthesizing a clock frequency through the on-chip PLL, or using the internal ring oscillator. Six other global buffers provide high-fanout signals to CSL functions. The bus clock and the global buffers can optionally be stopped upon a breakpoint event and shut off during power-down mode.

Power management control provides selectable power-down options over internal functions. Furthermore, each PIO provides pin-by-pin power-down settings. An internal initialization boot ROM controls the start of initialization after power-on or after the reset pin is released. The primary purpose of the initialization boot ROM is to find the user's initialization data and code stored in the secondary boot code, usually held in external non-volatile memory. Initialization programs can be written to external flash via JTAG through the MSSIU interface.

Besides downloading initialization programs, the JTAG port offers nearly full access to the microcontroller, peripherals, and CSL functions to aid in debugging. The JTAG interface can become a bus master on the internal CSI bus. During the system application, the JTAG unit provides observability of the system with minimal system impact.

Debugging the application software is done using a set of debugging tools. A CSI breakpoint unit 150 monitors both the CPU local bus or the CSI bus. Upon a predefined set of conditions, the breakpoint unit can halt or

interrupt the execution of the application program. Via JTAG control, the user can then control the behavior of the system.

Tracing of the local CPU bus or the CSI bus is also available through the breakpoint unit and the scratchpad memory. In trace mode, the upper 8K of the scratchpad memory is reclaimed for tracing. Different trace modes are supported and are described in more details in a later chapter.

Using JTAG and the breakpoint unit, it is possible to single step the CPU, the CSL system clock and transactions on the CSI independently of each other. Single stepping on the CSI is done at the transaction level and allows a single transaction to propagate through the CSI pipeline until completed. Multiple memory aliasing for a configurable system on a chip extends the bottom of memory mapping to other memories, and also serves the configuration requirements of a configurable system-on-chip.

In the configurable system-on-chip shown in **Figure 1**, the system contains an internal ROM, an external non-volatile memory (FLASH), an internal SRAM and some external SDRAM. The internal ROM is a non-volatile memory. Upon device reset, the memories are aliased at address 0. They overlay each other with the ROM having the highest priority, followed by the FLASH, then the internal scratchpad SRAM and finally the SDRAM. In addition to their aliases, all these memories have an image on top of the memory map. The ROM alias contains the necessary minimal primary

initialization code. The external non-volatile memory alias is used by the RTOS and by the secondary initialization code. The SDRAM could be used to run the user application. The user program in the FLASH memory could copy and decompress itself into SDRAM. The external FLASH could be used for configuration and to hold a compressed version of the user application.

Figure 2 shows an embodiment of a method for multiple memory aliasing. Upon device reset 205, the CPU starts executing from the internal ROM and searches for valid secondary initialization code, 210. Once a valid configuration program is found in the external non-volatile memory, the code branches out to the ROM image at the top of the memory, 215, disables the ROM alias, 220, and then jumps to the secondary initialization routine, 225. In one embodiment, the secondary initialization code resides at the top of the external FLASH along with the RTOS and user application code at the bottom of the external FLASH. The secondary initialization code configures the device (hard and soft peripherals), 230. After the initialization sequence is completed, the control is released to the RTOS or user application by resetting the CPU, 235. It then starts executing at address 0 from the bottom of the external non-volatile memory, 240. At this point, there is no more difference between a regular processing system and the configurable system-on-chip. The application gets setup the same way it would have been if the system had been fixed all along, 245. The setup can copy the critical routines into the internal

SRAM and a copy of the code in the external SDRAM for maximum performance, after which it can disable the external non-volatile memory alias, 250. All the re-mapping control can be done through control registers.

Figures 3A, 3B and **3C** illustrate the different states of the memory map throughout the setup of the configurable system-on-chip. **Figure 3A** shows the first phase of the memory map upon device reset. **Figure 3B** shows the memory map during secondary initialization and during the start of the user application. **Figure 3C** shows a possible configuration during normal operation of the system.

The multiple memory aliasing for a configurable system on a chip allows the internal ROM to be added to the ARM system memory map, which traditionally consisted of only external FLASH, SRAM and SDRAM. The multiple memory aliasing allows all types of memories to be aliased to the bottom of the memory map. The multiple memory aliasing also enables programmable aliases for all types of memories. Also, multiple memory aliasing allows programmable priority of the aliasing. A user can program the priority of the external memories at the bottom of memory to satisfy different application requirements. For example, if the user wants to keep the FLASH alias at the bottom of memory, but have the scratchpad on top.

These and other embodiments of multiple memory aliasing may be realized in accordance with these teachings and it should be evident that

various modifications and changes may be made in these teachings without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than restrictive sense and the invention measured only in terms of the claims.